

REMARKS/ARGUMENTS

This Amendment is in response to the Final Office Action dated April 16, 2008. Claims 1-8 and 25-32 are pending. Claims 1-8 and 25-32 are rejected. Claims 1, 3-8, and 25-32 have been amended. Accordingly, claims 1-8 and 25-32 remain pending in the present application.

Specification and Drawings

The specification has been amended to correct typographical errors and to correct duplicate reference numbers in Figures 2 and 3. Figures 2 and 3 have been amended to be consistent with the amended specification.

Rejection under 35 USC 102(e)

Claims 1-8 and 25-32 are rejected under 35 USC 102(e) as being anticipated by Bakalash et al. (2002/0029207). Applicant respectfully disagrees as to the claims as amended.

The invention, as recited in amended independent claims 1 and 25, provide a method and code with instructions for aggregating a plurality of entries in a table in a database management system into an aggregated entry in the table or another table in the database management system. The aggregating comprises making the aggregated entry. The aggregated entry represents the plurality of entries. The entry also includes a field whose value is a representation of a set of individual members. The individual members are derived from values contained in entries belonging to the plurality of entries. The representation specifies the individual members of the set. Thus, the invention preserves the individual members in the field of the aggregated entry by specifying the individual members in the representation.

In contrast, Bakalash discloses aggregation into a summary, consolidation, or compilation of the individual members as logical subtotals and totals along all dimensions of the MDDB. (paragraph 29) Thus, the individual members are not preserved in Bakalash's aggregated entry. Specifically, Bakalash discloses that base data is loaded into a multidimensional database (MDDB) from a variety of operational systems within an enterprise. The base or atomic data is then aggregated along the orthogonal dimensions of the MDDB and fill the array structure thereof. For example, revenues figures for all retail stores in a particular state (i.e. New York) would be *added*

together to fill the state level cells in the MDDb. (paragraph 25 and 27)

In the context of relational databases, Bakalash discloses that in a star schema, the results of aggregation are summary tables. FIGS. 18A and 18B illustrate a multi-dimensional relational database using a star schema and summary tables. In this example, the summary tables are generated over the “time” dimension storing aggregated data for “month”, “quarter” and “year” time periods. Summary tables are in essence additional fact tables, of higher levels. (paragraph 74) However, Bakalash discloses that summary tables do not provide a mechanism that allows efficient drill down to view the raw data that makes up the summary table – typically a table scan of one or more large tables is required. (paragraph 77) This is because in Bakalash, unlike the recited invention, the raw data is not preserved or specified in the aggregated entry.

This difference between the recited invention and Bakalash is significant as it impacts the efficiency of analysis of the aggregated data. With the recited invention, since the aggregated entry preserves or specifies the individual members of the set, the depth of hierarchy is reduced as compared to Bakalash. The recited aggregated entry also makes it easier to drill down to view the raw data as compared to Bakalash. With the recited invention, the requirement for a table scan of one or more large tables is reduced. (Please see also the examples set forth in the specification at p. 11, line 30 through p. 13, line 22.)

For the above reasons, Bakalash does not teach or suggest the aggregated entry recited in amended independent claims 1 and 25. Claims 1 and 25 are thus allowable over Bakalash. Dependent claims 2-8 and 26-32 are also allowable over Bakalash because they depend upon these allowable base claims.

Patentability of the dependent claims in their own rights

Additionally, the dependent claims set forth limitations which are not disclosed in Bakalash, and are consequently patentable in their own rights over the references.

Claims 2 and 26 recite the further step/instruction: deleting the plurality of entries represented by the aggregated entry. In her rejection, Examiner refers Applicants to paragraphs [0216] and [0258]. [0216] refers to items of a "work list" that are deleted after they are processed. [0214] makes it clear that the items on the work list are copies of items on an ordered list of items; the deletion of an item on the work list thus has no

effect on the ordered list of items or on the actual sources of the ordered list. [0258] discusses FIG. 22, which discloses an embodiment in which an RDBMS is a component of a data warehouse system. There is absolutely nothing in [0258] concerning deletion of data from tables. Thus, neither [0216] nor [0258] discloses the added limitation and claims 2 and 26 are patentable in their own rights over the references.

Claims 3-6 and 27-30 set forth further limitations concerning claim 1 and claim 26's limitation of the "representation specifying individual members of the set". Since Bakalash does not disclose the recited representation, as set forth above, the reference cannot disclose the further limitations of the representation and these claims as well are patentable in their own rights over the reference.

Conclusion

Applicant respectfully requests reconsideration and passage to issue of claims 1-8 and 25-32 as now presented. Should any unresolved issues remain, Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Please charge any additional fees required for the amendment or refund any overpayments to deposit account number 501315.

Respectfully submitted,

/Michele Baillie/
Michele Baillie
Gordon E. Nelson, Patent Attorney, P.C.
57 Central St., P.O. Box 782
Rowley, MA 01969
Registration number 44,875
Voice: (978) 948-7632
Fax: (866)-723-0359
6/12/2008
Date